DETAILED ACTION

This action is in response to the amendment filed on 12/29/2011.

Claims 14-19 and 33-38 are cancelled by the Applicant.

Claims 1, 20 and 39-50 are amended by the Applicant.

Claims 1-12, 20-31, 39-49 are allowed.

Examiner's Amendment

An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with KEITH MULLERVY [62,382] on February 2nd, 2012. Examiner's amendment is necessitated to overcome the rejection and further clarify the claimed invention. Amendments to all independent claims were discussed. The purpose of the amendment is to clarify the invention and to patentably distinguish the invention from the prior art of the record.

In the claims:

Please cancel claims 13, 32 and 50.

Please amend claims 1, 20 and 39 as follows.

1. (Currently Amended) A method of computing comprising:

reading, by an execution engine, a data processing representation having code sections with code statements of at least a first and a second programming language;

recognizing, by the execution engine, a first code section with at least code statements of a first programming language;

invoking, by the execution engine, a first code statement processing unit of the first programming language to process the first code section;

invoking, by the execution engine, a second code statement processing unit of a second programming language to process a code statement, when the first code statement processing unit locates a code statement of the second programming language within the first code section, and invokes the execution engine recursively;

recognizing, by the execution engine, a second code section with at least code statements of the second programming language;

invoking, by the execution engine, the second code statement processing unit of the second programming language to process the second code section; and

invoking, by the execution engine, the first code statement processing unit of the first programming language to process a code statement, when the second code statement processing unit locates a code statement of the first programming language within the second code section, and invokes the execution engine recursively;

wherein the code statement of the second programming language within the first code section, and the code statement of the first programming language within the second code section, are both within the data processing representation; and

recognizing a header section of a selected one of the first and the second

programming language;

recognizing a declare statement within the header section, enumerating one or more instance variables; and

instantiating the enumerated one or more instance variables for use within code sections with at least statements of the selected first and second programming language.

13. (Cancelled)

20. (Currently Amended) An apparatus comprising:

at least one storage unit having stored thereon programming instructions designed to instantiate an execution engine to enable the apparatus to

read, by the execution engine, a data processing representation having code sections with code statements of at least a first and a second programming language,

recognize, by the execution engine, a first code section with code statements of at least the first programming language,

invoke, by the execution engine, a first code statement processing unit of the first programming language to process the first code section,

invoke, by the execution engine, a second code statement processing unit of a second programming language to process a code statement, when the first code statement processing unit locates a code statement of the second programming language within the first code section, and invokes the execution engine recursively;

recognize, by the execution engine, a second code section with code statements of at least the second programming language,

invoke, by the execution engine, a second code statement processing unit of the second programming language to process the second code section,

invoke, by the execution engine, the first code statement processing unit of the first programming language to process a code statement, when the second code statement processing unit locates a code statement of the first programming language within the second code section, and invokes the execution engine recursively;

wherein the code statement of the second programming language within the first code section, and the code statement of the first programming language within the second code section, are both within the data processing representation; and

wherein said programming instructions are further designed to enable the apparatus to recognize a header section of a selected one of the first and the second programming language;

recognize a declare statement within the header section, enumerating one or more instance variables; and

instantiate the enumerated one or more instance variables for use code sections with at least code statements of the selected one of the first and the second programming language; and

at least one processor coupled to said at least one storage unit to execute said programming instructions.

32. (Cancelled)

39. (Currently Amended) A non-transitory computer-readable medium having

instructions stored thereon that, when executed by a processor, cause the processor to

implement an execution engine, the instructions comprising:

reading a data processing representation having code sections with code statements of at

least a first and a second programming language;

recognizing a first code section with at least code statements of a first programming

language;

invoking a first code statement processing unit of the first programming language to

process the first code section;

invoking a second code statement processing unit of a second programming language to

process a code statement, when the first code statement processing unit locates a code

statement of the second programming language within the first code section, and invokes the

execution engine recursively;

recognizing a second code section with at least code statements of the second

programming language;

invoking the second code statement processing unit of the second programming language

to process the second code section; and

invoking the first code statement processing unit of the first programming language to

process a code statement, when the second code statement processing unit locates a code

statement of the first programming language within the second code section, and invokes the execution engine recursively;

wherein the code statement of the second programming language within the first code section, and the code statement of the first programming language within the second code section, are both within the data processing representation; and

wherein the instructions further comprise

recognizing a header section of a selected one of the first and the second programming language;

recognizing a declare statement within the header section, enumerating one or more instance variables; and

instantiating the enumerated one or more instance variables for use within code sections with at least statements of the selected first and second programming language.

50. (Cancelled)

--END--

Reasons for Allowance

The following is an examiner's statement of reasons for allowance:

The cited prior art taken alone or in combination fail to teach, in combination with the other claimed limitations, ...*invoking, by the execution engine, a second code statement processing unit of a second programming language to process a code statement, when the first code*

statement processing unit locates a code statement of the second programming language within the first code section, and invokes the execution engine recursively; recognizing, by the execution engine, a second code section with at least code statements of the second programming language; invoking, by the execution engine, the second code statement processing unit of the second programming language to process the second code section; and invoking, by the execution engine, the first code statement processing unit of the first programming language to process a code statement, when the second code statement processing unit locates a code statement of the first programming language within the second code section, and invokes the execution engine recursively; wherein the code statement of the second programming language within the first code section, and the code statement of the first programming language within the second code section, are both within the data processing representation; and recognizing a header section of a selected one of the first and the second programming language; recognizing a declare statement within the header section, enumerating one or more instance variables; and instantiating the enumerated one or more instance variables for use within code sections with at least statements of the selected first and second programming language as recited in the independent claims 1, 20, and 39.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner

should be directed to Satish Rampuria whose telephone number is (571) 272-3732. The

examiner can normally be reached on 8:30 am to 5:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor,

Wei Y. Zhen can be reached on (571) 272-3708. Any inquiry of a general nature or relating to

the status of this application should be directed to the TC 2100 Group receptionist: 571-272-

2100.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Satish Rampuria/
Examiner, Art Unit 2191
/WEI ZHEN/
Supervisory Patent Examiner, Art Unit
2191